Integrating Vision and Robotics into the Computer Science Curriculum

Grzegorz Cielniak[1], Nicola Bellotto[1] and Tom Duckett[1]

[1] Centre for Vision and Robotics Research, School of Computer Science,
University of Lincoln, Brayford Pool, LN6 7TS Lincoln, UK
{gcielniak, nbellotto, tduckett}@lincoln.ac.uk

Abstract. This paper describes our efforts in integrating Robotics education into the undergraduate Computer Science curriculum. Our approach delivers Mobile Robotics together with the closely related field of Computer Vision and is directly linked to the research conducted at our institution. The paper describes the most relevant details related to the module content and assessment strategy, paying particular attention to the practical sessions using Rovio mobile webcams. We discuss the specific choices made with regard to the mobile platform, software libraries and lab environment. We also present a detailed qualitative and quantitative analysis, including the correlation between student engagement and performance, and discuss the outcomes of this experience.

Keywords: robotics education, Rovio, programming, Student as Producer.

1 Introduction

This paper describes the integration of Robotics education into the undergraduate curriculum at the University of Lincoln, UK. Rather than teaching Robotics in isolation, the subject is tightly integrated with other disciplines, in particular Computer Vision, as well as the ongoing activities of our research centre. Our approach to Robotics teaching features a holistic combination of theory and practical work, including programming of vision-guided mobile robots in an open-ended assignment loosely based on the popular RoboCup football tournament. This strategy also reflects the current policies of our institution on research-informed teaching and the "Student as Producer", a university-wide initiative supported by the UK Higher Education Academy. Student as Producer "restates the meaning and purpose of higher education by reconnecting the core activities of universities, i.e., research and teaching, in a way that consolidates and substantiates the values of academic life" [1].

Here Robotics is delivered primarily in a study module called Computer Vision and Robotics. The module is targeted at our 3rd (i.e. final) year undergraduate students due to its advanced content, assuming significant programming skills and having a focus on mobile robots as complete systems, as well as their components. The module is compulsory for students on the Computer Science programme, and optional for students studying other programmes at our School (Games Computing, Computer Information Systems, and Web Technology). There are several relevant modules that students undertake in earlier years of study which provide the necessary background in programming and basic knowledge of Artificial Intelligence (AI), including

Proceedings of 3rd International Workshop
Teaching Robotics, Teaching with Robotics
Integrating Robotics in School Curriculum
Riva del Garda (Trento, Italy) April 20, 2012
ISBN 978-88-95872-05-6
pp. 57-66

Software Development in year 1, and Advanced Software Development and AI in year 2. In addition, students have the option to study robotics as part of their self-guided Individual Study Project in year 3, in parallel with this taught module.

Table 1 Main topics in Semester A (Computer Vision) and Semester B (Mobile Robotics), with indicative coverage by assessment (● = high, ○ = low), detailed in Table 2.

Lecture	Topic	Assessment		
		C1	C2	C3
A1	Introduction to Computer Vision	○		○
A2	Introduction to Linear Algebra and MATLAB	●		
A3	An Overview of Pattern Recognition	●		○
A4	Spatial Processing and Filtering	●	○	○
A5	Colour Image Processing	●	○	○
A6	Morphological Image Processing	○	○	●
A7	Image Segmentation I		○	●
A8	Image Segmentation II		○	●
A9	Image Representation and Description			●
A10	Pattern Classification			●
B1	Introduction to Robotics		○	○
B2	Robot Programming in C#		●	
B3	Actuators and Sensors		●	●
B4	Robot Vision		●	○
B5	Robot Control		●	●
B6	Robot Behaviours		●	●
B7	Control Architectures I		○	●
B8	Control Architectures II		○	●
B9	Navigation Strategies			●
B10	Robotic Map Building			●

Table 2 The three assessments for the module including weightings.

Assessment	Weighting	Description	Semester
C1	30%	Digital Image Processing in MATLAB	A
C2	30%	Vision-based Robot Control in C#	B
C3	40%	Written Examination (3 hours)	A + B

2 Module description

'Computer Vision and Robotics' consists of two distinctive parts: Computer Vision fundamentals are covered in the first semester ('A') and Robotics is covered in the second semester ('B') – see Table 1 for a detailed list of topics. The Computer Vision part has a particular emphasis on pattern recognition applications, hence the early overview on this topic, and otherwise follows a fairly standard selection of topics in digital image processing from a well-known textbook [2]. The weekly lectures (1 hour each), with additional supporting materials, are accompanied by a weekly 2-hour workshop in which the students learn to program in MATLAB using the Image Processing Toolbox (IPT) and the corresponding textbook [3]. Many techniques and concepts learnt in the Computer Vision part are directly relevant and applicable to the Robotics content. Indeed, robot vision is a primary focus of the practical sessions and

Proceedings of 3rd International Workshop
Teaching Robotics, Teaching with Robotics
Integrating Robotics in School Curriculum
Riva del Garda (Trento, Italy) April 20, 2012
ISBN 978-88-95872-05-6
pp. 57-66

the assignment task in the second semester. Each semester is individually assessed through a practical assignment (see Section 3 for more details) and there is a final written examination covering the theoretical content of both semesters.

The Robotics part of the module includes 12 weeks of lecture delivery (comprising 10 major topics plus 2 weeks for supporting activities) and practical workshop sessions. The topics covered include the main challenges of robotics, robotic components, relevant software libraries, robot vision and control, robotic architectures, navigation strategies and map building. The first set of lectures provides many practical examples related to the robotic platform used so that the students can experience a direct link between theory and practice. The material covered also includes examples from our own research to illustrate fundamental problems of perception (e.g. people detection) and control (e.g. navigation), though we are careful to focus mainly on "textbook" science where possible [4,5,6].

2.1 Practical sessions

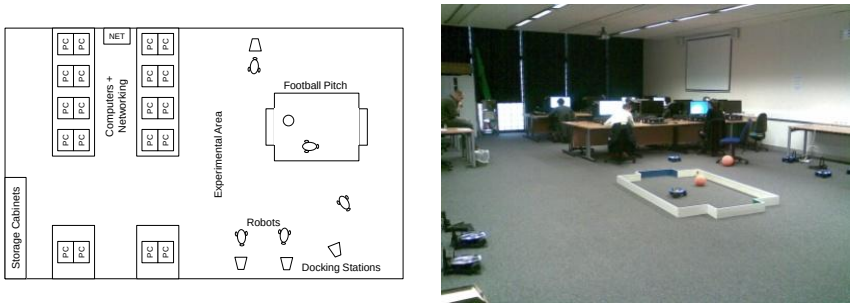

Figure 1 The layout of the computer lab used for practical sessions (*left*) and a snapshot taken during one of the sessions (*right*).

The Robotics part of the module features 12 two-hour long workshop sessions where the students have access to the robotic equipment. The sessions take place in a dedicated computer lab with storage facilities and necessary space for experimentation with the robots (see Fig. 1). With this arrangement, all frequent tasks such as unpacking, setting-up, charging, etc. can be carried out efficiently and smoothly without limiting the amount of time dedicated for interaction and work with the robots. In addition, the students can access the lab outside the workshop hours when the lab is not reserved for other activities. This arrangement appeared to be very popular last year, especially towards the assignment hand-in date (!).

The first four workshop sessions are designed as introductory tutorials where the students learn about the basic components of a robot, relevant software libraries and programming principles. The tasks include writing own object detection algorithms, visual feedback controllers, implementing simple behaviours or using Finite State Machines for sequencing more complex behaviours. The tutorial tasks were designed to provide a solid base and all necessary components needed for developing the following assignment task (see Section 3.2 for more details). The preparation of these teaching materials was assisted by a Masters' student who had studied the module in the previous academic year, as a 10-week summer project supported by the University

Proceedings of 3rd International Workshop
Teaching Robotics, Teaching with Robotics
Integrating Robotics in School Curriculum
Riva del Garda (Trento, Italy) April 20, 2012
ISBN 978-88-95872-05-6
pp. 57-66

in connection with the Student as Producer [1]. Our policy of research-informed teaching is also aided by the part-time employment of current PhD students in Vision and Robotics as demonstrators during the workshop sessions.

2.2 Choice of Robot Platform

We had to make a number of choices regarding the platform and software libraries. Our students are exposed mostly to programming in C# in their previous years of study and therefore it was decided to adopt this programming language for the practical sessions. The students are provided with a simple software wrapper directly implementing robot commands as specified by the Rovio API document [7]. An alternative would be to rely on existing robotic suites such as Stage/Player [8], ROS [9] or Microsoft Robotics Developer Studio [10]; however, such a choice is perhaps more preferable for larger scale, long-term projects as it requires substantial effort and time to become familiar with these frameworks. At the same time, the students learn what to expect from a commercial platform in terms of software support and how to extend the functionality to meet their own needs. The wrapper library [11] was developed following object-oriented programming guidelines so that the students could apply the concepts learnt in previous years to a real physical platform.

The recommended image processing library is AForge [12], which provides a well structured and documented functionality including a variety of image filters, object detection algorithms, video streaming support, etc. Ideally, OpenCV [13] would be preferable due to its overall functionality, maturity and support. However, the existing C# wrappers are inefficient and rely on non-safe use of the language that could hinder the smoothness of the learning process.

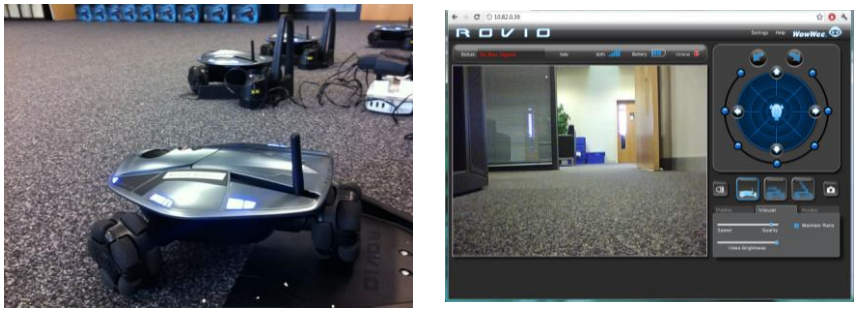

Figure 2 Rovio, the mobile robot (*left*) and the robot's web interface (*right*).

Rovio by WowWee [14] is an affordable mobile platform equipped with a set of sensors including a colour camera mounted on a moving head, odometry, infrared global navigation sensor and an infrared obstacle detector (see Fig. 2). The omni-directional drive configuration with omni-directional wheels enables holonomic movement in all directions. The communication with the robot is realised through wireless Ethernet: the onboard computer (ARM-based architecture) runs a web-server that accepts requests from and sends information to the remote PC which can be programmed to realise different custom behaviours.

There are several alternative robotic platforms that are popular in delivering Robotics courses, including Roomba and LEGO NXT. Our team had previous

Proceedings of 3rd International Workshop
Teaching Robotics, Teaching with Robotics
Integrating Robotics in School Curriculum
Riva del Garda (Trento, Italy) April 20, 2012
ISBN 978-88-95872-05-6
pp. 57-66

experience with these robots (using these robots for teaching the same module in the previous year), and based on this we prepared a subjective comparison of the popular robotic platforms presented in Table 3.

Table 3: A comparison of popular and affordable robotic platforms used for teaching Robotics in the classroom (support: ● = good, ◦ = fair, ○ = poor).

Feature	Rovio	Roomba	NXT
Affordability	●	●	◦
Maintenance/charging	●	◦	○
Set-up difficulty	●	◦	◦
Documentation	◦	◦	◦
Community Support	◦	○	●
Software components/libraries	●	●	●
Quality of components	◦	◦	●
Vision support	●	○	○

Whilst Roomba and LEGO NXT possess many attractive features (e.g. reconfigurable, direct control of wheels, etc.) which make these platforms very popular at other institutions (e.g. [15,16]), there are several reasons that make Rovio an ideal platform for delivery of Robotics in our context. First, the robot is equipped with a colour camera that is directly accessible from the software and can be used as a primary source of information in the workshop tasks. Other platforms (Roomba, NXT) require additional hardware components and solutions to enable video streaming. The platform is affordable – currently, it costs less than a single Roomba robot or a standard Lego NXT set. This makes it possible to purchase a large number of units that can be used independently by individual students in a relatively large groups – this is very important for maintaining student engagement in the workshops. This also makes Rovio an expendable platform – a broken or faulty unit can be easily replaced. A simple and sturdy design together with a recharging station result in a straightforward and easily maintained set-up that minimises the time overhead for the preparatory tasks required before each session. This is a very important consideration in the longer term and for larger groups of students. The robot features a convenient communication interface through wireless Ethernet, enabling easy setup in existing computer laboratories (no drivers required, no additional dongles, connections, etc.) and a straightforward API that simplifies development of the students' own robot behaviours – there are minimal dependencies on other software libraries. Each Rovio robot in the lab is assigned its own IP address, and can be connected to directly and simply through an ordinary web browser (an important consideration when trying to work with large groups of students, with one robot per student).

The frequent and intensive use of the Rovio robots in our module also revealed some limitations that might have not been obvious from the very beginning. For example: the odometry sensor is unreliable, the global navigation sensor is unusable in a multi-robot environment, the camera image has low quality, there can be connection and bandwidth problems which are partially caused by the existing network facilities, etc. On the other hand, these limitations helped the students to understand some of the fundamental problems in robotics, networking, real-time control, sensor noise, interaction of multiple complex systems, etc.

Proceedings of 3rd International Workshop
Teaching Robotics, Teaching with Robotics
Integrating Robotics in School Curriculum
Riva del Garda (Trento, Italy) April 20, 2012
ISBN 978-88-95872-05-6
pp. 57-66

3 Assessment

The assessment strategy consists of the three separate components, including theoretical and practical exercises, in the form of written documents or electronic files (i.e. program source code). Details of the topics covered by the assignments are shown in Table 1. The main objective of the first and second components, "Digital Image Processing in MATLAB" (C1) and "Vision-based Robot Control" (C2), is to assess student work during Semester A and B, respectively; the marks for these two components reflect mostly the performance of the developed systems. The last component (C3) is the final (closed-book) written exam, designed to assess the student comprehension of theoretical topics from the whole academic year. Each component carries a weighting, highlighted in Table 2, which is used at the end of year to compute the overall module mark.

3.1 Digital Image Processing in MATLAB (C1)

The scope of this component is to assess knowledge and understanding on different aspects of computer vision, related in particular to image processing. This component is subdivided into three workshop tasks with increasing level of difficulty, assessed at regular intervals through the semester. The three workshop tasks are titled "Binary Images & Introduction to Pattern Recognition", "Intensity Transforms & Spatial Filtering" and "Colour Image Processing & Face Detection". The deliverables for each task consist of a brief report describing the approach used to solve the problem and the obtained results, accompanied by the corresponding MATLAB source code. The latter is further demonstrated in a follow-up workshop session, where students are required to answer specific questions about their own MATLAB implementations.

3.2 Vision-based Robot Control (C2)

The aim of this assignment is to evaluate competence in two major learning outcomes, namely the application of computer vision techniques to solve practical problems and the application of AI control methods to mobile robotics. The assignment, which builds upon computer vision expertise from the first semester and some knowledge of AI from the previous year, is inspired by the RoboCup Middle-Size League competition [17].

The students are asked to design and develop a simplified version of robotic football. The RoboCup scenario is acting in this case as a micro-world [18] in which there are well understood objectives and requirements, as well as potential for experimentation and discovery. The students can select one type of player from a given list, comprising striker, midfielder, defender and goal keeper. The game features a uniformly coloured ball and a game field consisting of a rectangular enclosure with distinctively coloured goals installed in the computer lab (see Fig. 1).

The minimum requirement for player functionality includes a ball searching behaviour and a striking/defending behaviour, depending on the player type. Extra credit is given for developing additional components including, but not limited to: 1) enhanced object detection system for learning object appearance or using multiple cues; 2) behaviour coordination for deriving sophisticated game strategies and

Proceedings of 3rd International Workshop
Teaching Robotics, Teaching with Robotics
Integrating Robotics in School Curriculum
Riva del Garda (Trento, Italy) April 20, 2012
ISBN 978-88-95872-05-6
pp. 57-66

implementing awareness of other game objects, like goals and opponents; and 3) incorporation of the above information into the player behaviour.

The submission of this assignment should include a short technical report documenting the design, implementation and evaluation of the proposed functionality as well as developed source code. The report must reference any sources, such as textbooks or web-pages, which were used to help develop the solution. In addition, it is compulsory for the students to demonstrate their work during a separate workshop session after the submission.

3.3 Exam (C3)

The exam covers theoretical topics related to both parts of the module. It is particularly designed to evaluate students in their critical assessment of theoretical methods for Computer Vision and Mobile Robotics. In contrast to the previous assessments, the exam is taken at a pre-determined time and location, a few weeks after the end of Semester B, and must be completed within 3 hours.

The exam consists of a written test with questions on several Computer Vision (e.g. "Applied Pattern Recognition", "Image Segmentation" and "Morphological Image Processing") and Robotics topics (e.g. "Sensing and Control", "Control Architectures" and "Navigation"). The students have to answer four questions, two from Computer Vision and two from the Robotics section. There are no programming tasks in this final assessment. Instead, students are required to demonstrate their understanding of the systems and algorithms covered in the two semesters (see Table 1 for an overview), and to compare different solutions to Computer Vision and Robotics problems. In some cases, students are also asked to propose algorithmic solutions to specific problems, highlighting the necessary steps and tools.

4 Student performance

In the academic year 2010/11, the student cohort consisted of 18 Computer Science students for whom the module was compulsory, 4 Games Computing and 1 Web Technology student who chose the module as an option, giving 23 students in total.

4.1 Overall Results

Fig. 3 presents the comparative distribution of marks for each assessment component and sample correlation coefficients between marks for each pair of assessment components. While the practical sessions proved to be popular and the students received relatively good marks for both assignments (C1 and C2), the theoretical examination (C3) results were lower than expected – see the low correlation coefficients between the exam and both practical assignments. These figures might indicate a better engagement of our students in the practical part of the course, but perhaps also a wider problem of student comprehension of theoretical material observed across all programmes at our institution.

Proceedings of 3rd International Workshop
Teaching Robotics, Teaching with Robotics
Integrating Robotics in School Curriculum
Riva del Garda (Trento, Italy) April 20, 2012
ISBN 978-88-95872-05-6
pp. 57-66

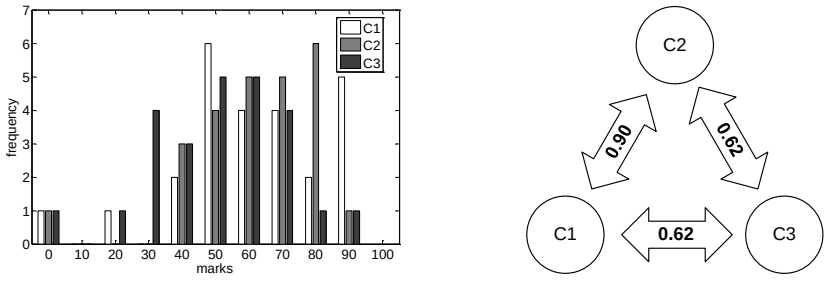

Figure 3 Distribution of marks obtained for different assessment components (*left*) and mark correlation between each pair of assignment component (*right*).

To further analyse the results, we also looked at attendance data as a basic indicator of student engagement (see Table 4). The results indicate that attendance of lectures and workshops showed a strong correlation with performance on the Computer Vision assignment (C1) but less so in the second semester. Perhaps surprisingly, the attendance data showed the lowest correlation with the marks obtained in the exam.

Table 4 Sample correlation coefficients between the marks obtained for the different components and the attendance recorded for the lectures and workshop sessions.

	C1	**C2**	**C3**
Lectures	0.82	0.63	0.52
Workshops	0.70	0.55	0.37

Table 5 Sample correlation coefficients between 'Computer Vision and Robotics' and other relevant modules (see Section 4.1 for full label details).

	CVR	**ASE**	**ISP**	**ASD**	**AI**	**SD**
SD	0.34	0.41	0.38	0.37	0.53	1.00
AI	0.49	0.40	0.49	0.62	1.00	
ASD	0.62	0.54	0.54	1.00		
ISP	0.87	0.87	1.00			
ASE	0.94	1.00				
CVR	1.00					

We also analysed the correlation between marks obtained in 'Computer Vision and Robotics' and other relevant modules taken by students during their course of study. Table 5 presents sample correlation coefficients in the form of a matrix between different modules including *3rd year modules:* Computer Vision and Robotics (CVR), Advanced Software Engineering (ASE) and Individual Study Project (ISP), *2nd year modules*: Advanced Software Development (ASD) and Artificial Intelligence (AI), and a *1st year module* Software Development (SD).

While it can be seen that the correlation between CVR and other 3rd year modules is stronger than with modules from earlier years, it is interesting to notice that CVR is ranked as the most correlated module with ASE, ISP and ASD, the modules where programming skills play a prominent role. CVR was ranked as the second-most correlated module with AI and the least-correlated with Software Development.

Proceedings of 3rd International Workshop
Teaching Robotics, Teaching with Robotics
Integrating Robotics in School Curriculum
Riva del Garda (Trento, Italy) April 20, 2012
ISBN 978-88-95872-05-6
pp. 57-66

4.2 Robotics Assignment – Observations

The robotics assignment had clearly defined minimum requirements but fairly open goals, which encouraged experimentation and exploration. This resulted in a number of exceptionally good submissions which included many features beyond the standard specifications. All top submissions (marks greater than 70%) had a clearly defined individual focus and explored different issues and directions. Some examples of the outstanding achievements presented by the students included:

- object detection:
  - a multi-stage image processing pipeline including cascaded segmentation in different colour spaces, morphological operators for noise filtering, use of edges as additional features;
  - histogram based tuning and learning of image filter parameters;
- control architectures:
  - a hybrid architecture combining reactive and deliberative approaches;
  - complex behaviour sequencing models (e.g. hierarchical FSM), a predictive ball search behaviour, multi-threading and synchronisation featuring customized threading queue mechanisms, off-line system development using pre-recorded data sets;
- system evaluation:
  - rigorous quantitative evaluation including switching time analysis of behavioural models and dedicated testing scenarios;
  - consideration of trade-offs (e.g. speed vs. accuracy) for robot controllers.

The above list of topics indicates that the students were able to apply knowledge learnt in the previous semester (i.e. object detection) but also other modules including AI, Software Development and Software Engineering.

Many of the listed techniques required the students to research sources other than the recommended reading material. On average, the top performing students referenced 2 items from the recommended reading list, 3 items that included other books, journal publications and conferences discovered through individual research, and 1 item referencing software or hardware.

5 Discussions and Conclusions

It has been pointed out previously that "educational robotics in general, is not precisely to teach learners to be robotics experts but to develop the essential competences to be successful in the present world" [15]. In our case, we believe that the module presented gives students vital expertise in areas that are otherwise not strongly covered in the "standard" Computer Science topics, such as dealing with complex systems at a systems (and systems of systems) level, combining hardware with sensing and control software, understanding the practicalities of real-time systems, understanding the inherent uncertainty in the real-world as perceived through sensors (sensor noise), applying "agile" programming methodologies in practice, etc.

We observed a very high engagement by the students, who spent a significant amount of time solving the robot football task required for the assignment. This resulted in a number of exceptional submissions with very original functionality beyond the assignment brief (e.g. threading queues, speech synthesis). While the

Proceedings of 3rd International Workshop
Teaching Robotics, Teaching with Robotics
Integrating Robotics in School Curriculum
Riva del Garda (Trento, Italy) April 20, 2012
ISBN 978-88-95872-05-6
pp. 57-66

evaluation part of the assignment was the part which students most often struggled with, many of the technical issues and platform shortcomings were identified by the students (connection and bandwidth problems, granularity of the movement commands, limited odometry, changing light conditions, etc.). The laboratory space encouraged cooperation, support and competition in developing individual solutions.

Our plans for future development of the module include: more flexible access to the lab, a common software repository to teach code maintenance and development in teams, extensions to the software environment, multi-robot scenarios, and a greater involvement in the Student as Producer initiative [1], including recruiting more student helpers and building stronger links with the student Computing Society. We also envisage a comparative study with other institutions delivering similar content.

Acknowledgment. We would especially like to thank our colleague Kevin Jacques for invaluable help with gathering the student data.

References

1. Neary, M., Winn, J.: The student as producer: reinventing the student experience in higher education. In: The future of higher education: policy, pedagogy and the student experience. Continuum, London, pp. 192--210 (2009)
2. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd edition, Prentice Hall (2002)
3. Gonzalez, R.C., Woods, R.E., Eddins, S.L.: Digital Image Processing using MATLAB, Prentice Hall (2004)
4. Bekey, G.A.: Autonomous robots. MIT Press (2005)
5. Siegwart, R., Nourbakhsh, I.R.: Introduction to autonomous mobile robots.MIT Press (2004)
6. Murphy, R. M.: Introduction to AI Robotics, MIT Press (2000)
7. Rovio API: http://www.wowweesupport.com/pdf/Rovio_API_Specifications_v1.2.pdf
8. Vaughan, R. T., Gerkey, B. P.: Reusable Robot Software and the Player/Stage Project. In Brugali, D. (eds.), Software Engineering for Experimental Robotics, Springer Berlin/Heidelberg (2007)
9. Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A. Y.: ROS: an open-source Robot Operating System. In: ICRA Workshop on Open Source Software, pp. 1--6 (2009)
10. Microsoft Robotics Developer Studio homepage: http://www.microsoft.com/robotics
11. RobotLib project website: http://robotlib.codeplex.com
12. LEGO NXT Mindstorms: http://mindstorms.lego.com
13. Bradski, G., Kaehler, A., Pisarevsky, V.: Learning-Based Computer Vision with Intel's Open Source Computer Vision Library. Intel Technology Journal, vol. 09(02), pp. 119--130 (2005)
14. WowWee Rovio: http://www.wowwee.com/en/products/tech/telepresence/rovio
15. Pittí, K., Curto, B., García, J., Moreno, V.: NXT Workshops: Constructionist Learning Experiences in Rural Areas. In: Proc. of SIMPAR2010 Conference, Workshop "Teaching robotics, teaching with robotics", Darmstadt, Nov. 15-16 (2010)
16. Menegatti, E., Moro, M.: Educational Robotics from high-school to Master of Science. In: Proc. of SIMPAR2010 Conference, Workshop "Teaching robotics, teaching with robotics", Darmstadt, Nov. 15-16 (2010)
17. RoboCup official Site: http://www.robocup.org/
18. Papert, S.: What is Logo? And Who Needs It? An essay. LCSI's book, Logo Philosophy and Implementation (1999)

Proceedings of 3rd International Workshop
Teaching Robotics, Teaching with Robotics
Integrating Robotics in School Curriculum
Riva del Garda (Trento, Italy) April 20, 2012
ISBN 978-88-95872-05-6
pp. 57-66